This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of:

randomly generating a plurality of **k** random odd numbers each providing a prime number candidate, including defining a length L for each of the plurality of **k** random numbers to be generated, and generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}; and

performing a plurality of **t** primality tests on each of said plurality of **k** randomly generated prime number whereby **(k x t)** tests are performed in parallel, each of the plurality of **(k x t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x t)** exponentiation units.

2. (Currently Amended) In a prime number generating system ~~as recited in claim 1~~ including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of:

randomly generating a plurality of **k** random odd numbers each providing a prime number candidate, wherein said plurality of **k** randomly generated numbers are expressed as $n_{0,0}$, $n_{1,0}$ ... $n_{(k-1)),0}$;~~, further comprising the steps of:~~

determining a plurality of **y** additional odd numbers based on each one of the randomly generated numbers $n_{0,0}$, $n_{1,0}$, ... $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates ($n_{0,1}$, $n_{0,2}$, ... $n_{0,y}$), ($n_{1,1}$, $n_{1,2}$, ... $n_{1,y}$), ...($n_{(k-1),1}$, $n_{(k-1),2}$, ... $n_{(k-1),y}$) thereby yielding a total number of **(k x (y+1))** prime number candidates; and

~~wherein said step of performing includes~~ performing **t** primality tests on each of said total number of **(k x (y+1))** prime number candidates, each of the plurality of **(k x (y+1) x**

Application No. 09/818,914
Amendment dated: 17 April 2006
Response to Office Action: March 02, 2006

t) primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x (y+1) x t)** exponentiation units, said exponentiation operations being performed in parallel by said plurality of **(k x (y+1) x t)** exponentiation units.

3. (original) In a prime number generating system as recited in claim 2 wherein each of said plurality of prime number values being searched for has a specified length, and wherein said plurality of y additional odd numbers defines an interval that is selected relative to said specified length.

4. (original) In a prime number generating system as recited in claim 2 wherein said step of determining a plurality of **y** additional odd numbers based on each one of the randomly generated numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ includes successively adding two to each of said randomly generated odd numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as $(n_{0,1}= n_{00}+ 2, n_{0,2}=n_{0,0}+ 4, \ldots n_{0,y} = n_{0,0} + (y\ 2))$, $(n_{1,1}= n_{1,0}+ 2, n_{1,2}=n_{1,0}+ 4, \ldots n_{1.y} = n_{1,0} + (y\ 2))$, $\ldots (n_{(k-1),1}= n_{(k-1),0}+ 2, n_{(k-1),2} =n_{(k-1,0}+ 4, \ldots n_{(k-1),y}= n_{(k-1),0} + (y\ 2))$.

5. (Currently Amended) In a prime number generating system as recited in claim [[~~1~~]] 2 wherein said exponentiation operations are performed by said associated exponentiation units simultaneously.

6. (Cancelled)

7. (Currently Amended) In a prime number generating system as recited in claim 1 further comprising the steps of:

sieving said **(k x (y+1))** prime number candidates by performing a small divisor test on each of said **(k x (y+1))** candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing said plurality of **t** primality tests on each of said sieved number **s** of candidates, each of primality tests including an associated

exponentiation operation executed by an associated one of a plurality of **st** of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of **st** exponentiation units~~-simultaneously~~.

8. (Currently Amended) In a prime number generating system as recited in claim 7 further comprising the steps of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of (a) each said prime number candidate (p_c) minus one ($p_c - 1$) and (b) said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing said plurality of primality tests.

9. (Currently Amended) In a prime number generating system as recited in claim 2 further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number s of candidates;

wherein said step of performing primality tests includes performing t primality tests on each of said sieved number **s** of candidates, each of the plurality of **s x t** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s x t** exponentiation units, said exponentiation operations being performed in parallel by said plurality of **s x t** exponentiation units.

10. (Cancelled)

11. (Currently Amended) In a prime number generating system as recited in claim 2 further comprising the step of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate

candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing primality tests includes performing an associated first one of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units in parallel in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units in parallel in order to eliminate further candidates revealed to be composite numbers.

12. (Currently Amended) In a prime number generating system as recited in claim [[~~1~~]] 2 wherein said step of performing primality tests includes performing a Fermat type primality test.

13. (Currently Amended) In a prime number generating system as recited in claim [[~~1~~]] 2 wherein said step of performing primality tests includes performing a Miller-Rabin type primality test.

14. (Original) In a prime number generating system as recited in claim [[~~1~~]] 2 wherein said step of randomly generating a plurality of **k** random odd numbers further includes:

defining a length L for each of the plurality of **k** random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

15. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of.

randomly generating at least one random odd number providing a prime number candidate;

determining a plurality of **y** additional odd numbers based on said at least one randomly generated odd number to provide **y** additional prime number candidates, thereby providing a total number of **y+1** candidates;

performing a plurality **t** primality tests on each of said **(y+1) x t** candidates, each of the **(y+1) x t** primality tests including an associated exponentiation operation executed by an associated one of **(y+1) x t** exponentiation units, said **(y+1) x t** exponentiation operations being performed in parallel by said associated **(y+1) x t** exponentiation units;

and further comprising the step of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of (a) each of said prime number candidates (p_c) minus one (p_c-1) and (b) a specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing said primality tests.

16. (Original) In a prime number generating system as recited in claim 15 wherein said at least one randomly generated odd number is expressed as $n_{0,0}$, and wherein said step of determining a plurality of y additional odd numbers based on said randomly generated odd number $n_{0,0}$ includes successively adding two to said randomly generated odd number $n_{0,0}$ to provide **y+1** additional prime number candidates expressed as ($n_{0,1}= n_{00}+ 2$, $n_{0,2}=n_{0,0}+ 4$, ... $n_{0,y} = n_{0,0} + (y\ 2)$).

17. (Cancelled)

18. (Previously Presented) In a prime number generating system as recited in claim 15 further comprising the step of sieving said **y+1** candidates by performing a small divisor

test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved numbers of candidates.

19. (Cancelled)

20. (Currently Amended) In a prime number generating system as recited in claim 15 further comprising the step of:

sieving said **y+1** candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing primality tests includes performing an associated first one of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units in parallel in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** additional primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** exponentiation units, the **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units in parallel in order to eliminate further candidates revealed to be composite numbers.

21. (Currently Amended) In a prime number generating system as recited in claim 15 wherein said step of [[2]] performing ~~at least one~~ primality tests includes performing a Fermat type primality test.

22. (Previously Presented) In a prime number generating system as recited in claim 15 wherein said step of performing primality tests includes performing a Miller-Rabin type primality test.

23. (Currently Amended) In a prime number generating system as recited in claim 15 wherein said step of randomly generating at least one random odd number further includes:

defining a length L for the or each of the ~~plurality of k random~~ odd numbers to be generated; and

generating the or each of said plurality of ~~k random~~ odd numbers in an interval between 2^L and 2^{L-1}.

24. (Previously Presented) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of:

randomly generating a plurality of random odd numbers providing prime number candidates;

sieving said candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

testing the primality of said sieved candidates by performing a first one of a plurality of **t** primality tests on said sieved number **s** of candidates, each of the plurality **s** of the first primality tests including an associated exponentiation operation executed by an associated one of a plurality **s** of the exponentiation units, said exponentiation operations being performed simultaneously by said plurality of **s** exponentiation units in order to eliminate candidates revealed to be composite numbers thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional ones of said t primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** exponentiation units, said **(r x (t-1))** exponentiation operations being simultaneously performed by said plurality of **(r x (t-1))** exponentiation units in order to eliminate further candidates revealed to be composite numbers.

25. (Cancelled)

26. (Currently Amended) In a prime number generating system as recited in claim 25 further including the step of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of (a) each said prime number candidates (p_c) minus one (p_c-1) and (b) said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing said first one of said primality tests.

27. (Cancelled)

28. (Previously Presented) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of:

randomly generating a plurality of **k** random odd numbers expressed as $n_{0,0}$, $n_{1,0}$,... $n_{(k-1),0}$, each said number providing a prime number candidate;

determining a plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{1,0}$,... $n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates ($n_{0,1}$, $n_{0,2'}$ ... $n_{0,y}$), ($n_{1,1}$, $n_{1,2}$, ... $n_{1,y}$), ...($n_{(k-1),1}$, $n_{(k-1,)2}$, ... $n_{(k-1),y}$) thereby yielding a total number of **(k x (y+1))** prime number candidates;

sieving said (k x (y+1)) prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates; and

performing at least one primality test on each of said sieved number s of candidates, each of the plurality of s primality tests including an associated exponentiation operation executed by an associated one of a plurality of s of the exponentiation units, said exponentiation operations being performed in parallel by said plurality of **s** exponentiation units in order to eliminate candidates revealed to be composite numbers by said primality test thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** exponentiation units, said **(r x (t-1))** exponentiation operations being performed by in parallel by said plurality of **(r x (t-1))** exponentiation units in order to eliminate further candidates revealed to be composite numbers.

29. (Original) In a prime number generating system as recited in claim 28 wherein said step of determining a plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{1,0},\ldots n_{(k-1),0}$ includes successively adding two to each of said randomly generated odd numbers $n_{1,0},\ldots n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as ($n_{0,1}= n_{00}+ 2$, $n_{0,2}=n_{0,0}+ 4$, ... $n_{0,y} = n_{0,0} + (y\ 2)$), ($n_{1,1}= n_{1,0}+ 2$, $n_{1,2}=n_{1,0}+ 4$, ... $n_{1.y} = n_{1,0} + (y\ 2)$), ... ($n_{(k-1),1}= n_{(k-1),0}+ 2$, $n_{(k-1),2} =n_{(k-1,0}+ 4$, ... $n_{(k-1),y}= n_{(k-1),0} + (y\ 2)$).

30. (Cancelled)

31. (Previously Presented) In a prime number generating system as recited in claim 28 wherein said step of performing said first primality test includes performing a Fermat type primality test.

32. (Previously Presented) In a prime number generating system as recited in claim 28 wherein said step of performing said first primality test includes performing a Miller-Rabin type primality test.

33. (Original) In a prime number generating system as recited in claim 28 wherein said step of randomly generating a plurality of k random odd numbers further includes:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

34. (Original) In a prime number generating system as recited in claim 28 wherein k is greater than or equal to 2.

35. (Currently Amended) In a prime number generating system as recited in claim 28 further comprising the steps of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said sieved prime number candidates for use in said cryptographic application by testing the relative primality of <u>(a)</u> each said sieved prime number candidate <u>(p_c)</u> minus one <u>(p_c-1)</u> and <u>(b)</u> said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing said first primality tests.

36. (Previously Presented) A prime number generating system for searching for a plurality of prime number values, comprising:

processing means operative to randomly generate a plurality of k random odd numbers each providing a prime number candidate, and to provide a plurality **t** sets of test parameters associated with a plurality of **t** primality tests to be performed on each one of said plurality of **k** randomly generated numbers, each **(k x t)** sets of said test parameters including said associated one of said plurality **k** randomly generated numbers and an associated one of a plurality of **t** base values; and

a plurality of exponentiation units each being communicatively coupled with said processing means, said plurality of exponentiation units includes a plurality of at least **(k x t)** exponentiation units each being responsive to an associated one of said **(k x t)** sets of test parameters, and operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said exponentiation units being

operative to perform said plurality of at least **(k x t)** exponentiation operations in parallel;

said processing means being responsive to said primality test result signals, and operative to process said test result signals for the purpose of eliminating randomly generated numbers declared to be composite in accordance with a search for prime number values.

37. (Cancelled)

38. (Original) A prime number generating system as recited in claim 36 wherein said processing means is further operative to sieve said prime number candidates by performing a small divisor test on each of said prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates.

39. (Previously Presented) A prime number generating system as recited in claim 36 wherein said plurality of k randomly generated odd numbers are expressed as $n_{0,0}$, $n_{1,0}$,... $n_{(k-1),0}$, and wherein:

said processing means is further operative to develop a plurality of **y** additional odd numbers based on each one of the randomly generated odd numbers $n_{0,0}$, $n_{1,0}$,... $n_{(k-1),0}$, to provide **(k x y)** additional prime number candidates ($n_{0,1}$, $n_{0,2}$, ... $n_{0,y}$), ($n_{1,1}$, $n_{1,2}$, ... $n_{1,y}$), ...($n_{(k-1),1}$, $n_{(k-1),2}$, ... $n_{(k-1),y}$) thereby yielding a total number of **(k x (y+1))** prime number candidates;

said plurality of exponentiation units includes a plurality of at least **(k x (y+1))** exponentiation units each being responsive to an associated one of said **(k x (y+1))** sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **(k x (y+1))** exponentiation

units being operative to perform the plurality of **(k x (y+1))** exponentiation operations in parallel.

40. (Original) A prime number generating system as recited in claim 39 wherein said processing means is operative to develop said plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{1,0},\ldots n_{(k-1),0}$, by successively adding two to each of said randomly generated odd numbers $n_{1,0},\ldots n_{(k-1),0}$, to provide **(k x y)** additional prime number candidates expressed as $(n_{0,1}= n_{00}+ 2, n_{0,2}=n_{0,0}+ 4, \ldots n_{0,y} = n_{0,0} + (y\ 2))$, $(n_{1,1}= n_{1,0}+ 2, n_{1,2}=n_{1,0}+ 4, \ldots n_{1.y} = n_{1,0} + (y\ 2))$, ... $(n_{(k-1),1}= n_{(k-1),0}+ 2, n_{(k-1),2} =n_{(k-1,0}+ 4, \ldots n_{(k-1),y}= n_{(k-1),0} + (y\ 2))$.

41. (Previously Presented) A prime number generating system as recited in claim 39 wherein:

said processing means is operative to provide a plurality of t sets of test parameters associated with a plurality of t primality tests to be performed on each one of said plurality of **(k x (y+1))** randomly generated numbers, each of the **(k x (y+1) x t)** sets of said test parameters including said associated one of said plurality of **(k x (y+1))** prime number candidates and an associated one of a plurality of **t** base values;

said plurality of exponentiation units includes a plurality of at least **(k x (y+1) x t)** exponentiation units each being responsive to an associated one of said **(k x (y+1) x t)** sets of said test parameters, and being operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **(k x (y+1) x t)** exponentiation units being operative to perform said plurality of **(k x (y+1) x t)** exponentiation operations in parallel.

42. (Original) A prime number generating system as recited in claim 36 wherein each of said primality tests is a Fermat type primality test.

43. (Original) A prime number generating system as recited in claim 36 wherein each of said primality tests is a Miller-Rabin type primality test.

44. (Original) A prime number generating system as recited in claim 36 wherein said processing means is operative to randomly generate said plurality of **k** random odd numbers by performing the steps of:

defining a length L for each of the plurality of **k** random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

45. (Previously Presented) A prime number generating system for searching for a plurality of prime number values, comprising:

processing means operative to randomly generate at least one random odd number providing a prime number candidate, and to determine a plurality of **y** additional odd numbers based on each of said at least one randomly generated odd number to provide **y** additional prime number candidates, thereby providing a total number of **y+1** candidates, said processing means also operative to provide a plurality of **t** sets of test parameters associated with a plurality **t** primality tests to be performed on each one of said **y+1** prime number candidates, each of the **((y+1(x t)** sets of test parameters including said associated one of said plurality of prime number candidates and an associated base value; and

a plurality of at least **((y+1) x t)** exponentiation units each communicatively coupled with said processing means, and each responsive to an associated one of said **((y+1) x t)** sets of test parameters, and operative to perform an exponentiation operation based on said associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **((y+1) x t)** exponentiation units being operative in parallel to perform said plurality of **((y+1) x t)** exponentiation;

said processing means being responsive to said primality test result signals, and operative to process said test result signals for the purpose of eliminating

randomly generated numbers declared to be composite in accordance with a search for prime number values.

46. (Previously Presented) A prime number generating system as recited in claim 45 wherein said at least one randomly generated odd number is expressed as $n_{0,0}$, and wherein said processing means is operative to determine said plurality of **y** additional odd numbers based on said randomly generated odd number by successively adding two to said randomly generated odd number $n_{0,0}$ to provide **(y+1)** additional prime number candidates expressed as ($n_{0,1}= n_{00}+ 2$, $n_{0,2}=n_{0,0}+ 4$, ... $n_{0,y} = n_{0,0} + (y 2)$).

47. (Cancelled)

48. (Original) A prime number generating system as recited in claim 45 wherein said processing means is further operative to sieve said prime number candidates by performing a small divisor test on each of said prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates.

49. (Previously Presented) A prime number generating system as recited in claim 45 wherein:

said processing means is operative to generate a plurality of **k** random odd numbers each providing a prime number candidate, and to determine a plurality of **y** additional odd numbers based on each of said **k** random odd numbers to provide **k x y** additional prime number candidates, thereby providing a total number of at least **k x (y+1)** candidates, said processing means also operative to provide **t** sets of test parameters associated with **t** primality tests to be performed on each one of said **k x (y+1)** prime number candidates, each said set of said test parameters including said associated prime number candidate and an associated base value; and

said plurality of exponentiation units includes a plurality of at least **(k x (y+1) x t)** exponentiation units each being responsive to an associated one of said **(k x (y+1) x t)** sets of said test parameters, and being operative to perform an exponentiation operation based on said

associated set of test parameters, and also operative to generate a primality test result signal declaring said associated prime number candidate to be either composite or prime with reference to said associated base value, said plurality of at least **(k x (y+1) x t)** exponentiation units being operative to perform said plurality of **(k x (y+1) x t)** exponentiation operations in parallel.

50. (Original) A prime number generating system as recited in claim 45 wherein each of said primality tests is a Fermat type primality test.

51. (Original) A prime number generating system as recited in claim 45 wherein each of said primality tests is a Miller-Rabin type primality test

52. (Currently Amended) A prime number generating system as recited in claim 49 wherein said processing means is operative to randomly generate said plurality of k random odd numbers by performing the steps of:

defining a length L for each of the plurality of k random numbers to be generated; and

generating each of said plurality of k random odd numbers in an interval between 2^L and 2^{L-1}.

53. (Cancelled)

54. (Currently Amended) A computer readable storage medium having stored thereon encoding instructions for executing a process of searching for a plurality of prime number values in a prime number generation system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, the process comprising the steps of:

randomly generating a plurality of **k** random odd numbers each providing a prime number candidate, said plurality of **k** randomly generated numbers are expressed as $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$; ~~and~~

Application No. 09/818,914
Amendment dated: 17 April 2006
Response to Office Action: March 02, 2006

determining a plurality of **y** additional odd numbers based on each one of the randomly generated numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates $(n_{0,1}, n_{0,2}, \ldots n_{0,y}), (n_{1,1}, n_{1,2}, \ldots n_{1,y}), \ldots (n_{(k-1),1}, n_{(k-1),2}, \ldots n_{(k-1),y})$ thereby yielding a total number of **(k x (y+1))** prime number candidates; and

performing a plurality of **t** primality tests on each of said ~~plurality of k randomly generated~~ total number of **(k x (y+1))** prime numbers whereby **(k x (y+1) x t)** tests are performed in parallel, each of the plurality of **(k x (y+1) x t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x (y+1) x t)** exponentiation units.

55. (Cancelled)

56. (Currently Amended) A computer readable storage medium as recited in claim [[~~55~~]] 54 wherein said step of determining a plurality of **y** additional odd numbers based on each one of the randomly generated numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ includes successively adding two to each of said randomly generated odd numbers $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$ to provide **(k x y)** additional prime number candidates expressed as $(n_{0,1} = n_{0,0} + 2, n_{0,2} = n_{0,0} + 4, \ldots n_{0,y} = n_{0,0} + (y\ 2)), (n_{1,1} = n_{1,0} + 2, n_{1,2} = n_{1,0} + 4, \ldots n_{1.y} = n_{1,0} + (y\ 2)), \ldots (n_{(k-1),1} = n_{(k-1),0} + 2, n_{(k-1),2} = n_{(k-1,0} + 4, \ldots n_{(k-1),y} = n_{(k-1),0} + (y\ 2))$.

57. (Cancelled)

58. (Cancelled)

59. (Cancelled)

60. (Currently Amended) A computer readable storage medium as recited in claim 54 [[55]] further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate

candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s x t** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s x** t exponentiation units, said exponentiation operations being performed by said plurality of **s x t** exponentiation units simultaneously.

61. (Cancelled)

62. (Currently Amended) A computer readable storage medium as recited in claim 54 [[55]] further comprising the steps of:

sieving said prime number candidates by performing a small divisor test on each of said number **(k x (y+1))** of prime number candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number **s** of candidates;

wherein said step of performing includes performing an associated first one of **t** primality tests on each of said sieved number **s** of candidates, each of the plurality of **s** first primality tests including an associated exponentiation operation executed by an associated one of a plurality of **s** of the exponentiation units, said first exponentiation operations being performed by said plurality of **s** exponentiation units in parallel in order to eliminate candidates revealed to be composite numbers by said first primality tests thereby yielding a remaining number **r** of candidates; and

performing a plurality of **t-1** additional primality tests on each of said remaining number r of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** exponentiation units, said **(r x (t-1))** exponentiation operations being performed by said plurality of **(r x (t-1))** exponentiation units in parallel in order to eliminate further candidates revealed to be composite numbers.

63. (Currently Amended) A computer readable storage medium as recited in claim 60 [[59]] further comprising the steps of:

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of (a) each said prime number candidate (p_c) minus one (p_c-1) and (b) said specified public exponent e, wherein said step of testing the suitability is performed prior to said step of performing said primality tests.

64. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of:

randomly generating a plurality of random odd numbers each providing a prime number candidate; and

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said prime number candidates for use in said cryptographic application by testing the relative primality of (a) each said prime number candidate (p_c) minus one (p_c-1) and (b) said specified public exponent e to yield a plurality of **k** suitable prime number candidates; and

performing a plurality of **t** primality tests on each of said plurality of **k** suitable prime number candidates whereby **(k x t)** tests are performed in parallel, each of the plurality of **(k x t)** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(k x t)** exponentiation units.

65. (Currently Amended) In a prime number generating system including a processing unit and a plurality of exponentiation units communicatively coupled with the processing unit, a process of searching for a plurality of prime number values, comprising the steps of:

randomly generating a plurality of k random odd numbers expressed as $n_{0,0}, n_{1,0}, \ldots n_{(k-1),0}$, each said number providing a prime number candidate;

determining a plurality of y additional odd numbers based on each one of the randomly generated odd numbers $n_{1,0}, \ldots n_{(k-1),0}$ to provide (k x y) additional prime number candidates

Application No. 09/818,914
Amendment dated: 17 April 2006
Response to Office Action: March 02, 2006

$(n_{0,1}, n_{0,2}, \ldots n_{0,y}), (n_{1,1}, n_{1,2}, \ldots n_{1,y}), \ldots (n_{(k-1),1}, n_{(k-1),2}, \ldots n_{(k-1),y})$ thereby yielding a total number of $(k \times (y+1))$ prime number candidates;

sieving said $(k \times (y+1))$ prime number candidates by performing a small divisor test on each of said candidates in order to eliminate candidates revealed to be composite numbers by said small divisor test thereby yielding a sieved number of candidates;

receiving a specified public exponent e associated with a cryptographic application;

testing the suitability of each of said sieved prime number candidates for use in said cryptographic application by testing the relative primality of (a) each said sieved prime number candidate (pc) minus one (pc-1) and (b) said specified public exponent e to yield **s** suitable sieved candidates; and

performing at least one primality test on each of said **s** suitable sieved candidates, each of the plurality of **s** primality tests including an associated exponentiation operation executed by an associated one of a plurality of s exponentiation units, said exponentiation operations being performed in parallel by said plurality of **s** exponentiation units in order to eliminate candidates revealed to be composite numbers by said primality test thereby yielding a remaining number **r** of candidates.

66. (Previously Presented) In a prime number generating system as recited in claim 65 further comprising the steps of:

performing a plurality of **t-1** additional primality tests on each of said remaining number **r** of candidates, each of the plurality of **(r x (t-1))** primality tests including an associated exponentiation operation executed by an associated one of a plurality of **(r x (t-1))** exponentiation units, said **(r x (t-1))** exponentiation operations being performed by in parallel by said plurality of **(r x (t-1))** exponentiation units in order to eliminate further candidates revealed to be composite numbers.